

**WIND TUNNEL MANAGEMENT AND RESOURCE OPTIMIZATION: A  
SYSTEMS MODELING APPROACH**

**Submitted to**

**NASA LANGLEY RESEARCH CENTER**

**by**

**Faculty Advisor: Derya A. Jacobs**

**Graduate Student: Curtis A. Aasen**

**Department of Engineering Management  
College of Engineering and Technology  
Old Dominion University**

## TABLE OF CONTENTS

	Chapter
1. INTRODUCTION .....	2
2. REVIEW OF PROJECT SCHEDULING AND MANAGEMENT .....	3
2.1. Project Evaluation and Review Technique .....	3
2.2. Stochastic-Resource Constrained Problem .....	5
3. IMPLEMENTATION OF AN INTEGRATED WIND TUNNEL PLANNING SYSTEM: THOERY OF CONSTRAINTS .....	5
4. CONCLUSIONS AND FURTHER RESEARCH.....	16

# **WIND TUNNEL MANAGEMENT AND RESOURCE OPTIMIZATION: A SYSTEMS MODELING APPROACH**

## **1. INTRODUCTION**

Time, money, and, personnel are becoming increasingly scarce resources within government agencies due to a reduction in funding and the desire to demonstrate responsible economic efficiency. The ability of an organization to plan and schedule resources effectively can provide the necessary leverage to improve productivity, provide continuous support to all projects, and insure flexibility in a rapidly changing environment. Without adequate internal controls the organization is forced to rely on external support, waste precious resources, and risk an inefficient response to change. Management systems must be developed and applied that strive to maximize the utility of existing resources in order to achieve the goal of “faster, cheaper, better”.

An area of concern within NASA Langley Research Center was the scheduling, planning, and resource management of the Wind Tunnel Enterprise operations. Nine wind tunnels make up the Enterprise. Prior to this research, these wind tunnel groups did not employ a rigorous or standardized management planning system. In addition, each wind tunnel unit operated from a position of autonomy, with little coordination of clients, resources, or project control.

For operating and planning purposes, each wind tunnel operating unit must balance inputs from a variety of sources. Although each unit is managed by individual Facility Operations groups, other stakeholders influence wind tunnel operations. These groups include, for example, the various researchers and clients who use the facility, the Facility System Engineering Division (FSED) tasked with wind tunnel repair and upgrade, the Langley Research Center (LaRC) Fabrication (FAB) group which fabricates repair parts and provides test model upkeep, the NASA and LARC Strategic Plans, and unscheduled use of the facilities by important clients. Expanding these influences horizontally through nine wind tunnel operations and vertically along the NASA management structure greatly increases the complexity of developing a model that can be used for successfully implementing a standardized management planning tool.

The objective of this study was to implement an Integrated Wind Tunnel Planning System to improve the operations within the aeronautics testing and research group, in particular Wind Tunnel Enterprise. The study included following steps:

- Conducted literature search and expert discussions (NASA and Old Dominion University faculty).
- Performed environmental scan of NASA Langley wind tunnel operations as foundation for problem definition.
- Established operation requirements and evaluation methodologies.
- Examined windtunnel operations to map out the common characteristics, critical components, and system structure.
- Reviewed and evaluated various project scheduling and management systems for implementation.
- Evaluated and implemented “*Theory of Constraints (TOC)*” project scheduling methodology at NASA Langley wind tunnel operations together with NASA staff.

## 2. REVIEW OF PROJECT SCHEDULING AND MANAGEMENT

The problems associated with project management and planning are not unique to NASA LaRC. Indeed, throughout industry and government the successful scheduling and control of projects has become increasingly important as pressures from shrinking resource availability, quicker time to market requirements, and greater task complexity continue to emerge. In order to address these problems, tools based on deterministic scheduling, queuing networks, and software engineering, have been researched and provided managers since 1950's. The use of heuristic techniques has played an important role in the development of the software tools. Which attempt to blend modeling and algorithmic sophistication.

Project scheduling and management are common to many different businesses and engineering environments. But whether the project is as large as the Boston Harbor Tunnel or something as seemingly simple as the redesign of the packaging for a tape dispenser, both planning and scheduling are profoundly important. Even on a small project, the number of possible courses of action and the number of ways to allocate resources quickly become overwhelming.

### 2.1. Project Evaluation and Review Technique

Many methods are used for project planning. Perhaps the most common planning tool is Project Evaluation and Review Technique (PERT). PERT specifies technological or other types of

precedence constraints among tasks in a project. Work breakdown structure (WBS) organizes tasks into functional groups. Other methods such as "four fields" focus on team members or project components rather than simply precedence relationships. Most project planning tools include more than one view of the project; a typical project planner will present a PERT chart to show precedence relationships, a WBS chart to show hierarchical or technological relationships, and some kind of temporal view, typically a Gantt chart, to illustrate estimated and actual start and completion times. In addition, many different resource usage reports are typically generated.

Scheduling methods typically start with a project plan then assign start times given temporal, precedence, or resource constraints. For example, one commonly used scheduling technique, the Critical Path Method (CPM), starts with a PERT chart then schedules tasks based on the precedence relationships and estimates of task duration in the PERT representation.

However, the CPM assumes infinite resources; if resources are not immediately available, they can be obtained at some cost. When resource constraints are added, the problem becomes much more difficult to solve. And in most cases, the optimal solution is unknown. This is almost always true as the number of tasks and number of resources increases. In addition, the CPM (and many other scheduling methods) assumes that tasks are atomic and cannot overlap. Standard PERT/CPM techniques are weak in this respect; a traditional PERT representation cannot model overlapping precedence relationships.

It is important to make a distinction between planning and scheduling. Using traditional tools, the PERT chart represents the project plan whereas the Gantt chart represents the project schedule. Often scheduling and planning are not done at the same time; scheduling is much easier once a plan has been established, and thus typically follows planning. As a result, planning and scheduling are often done iteratively (or in some cases, sequentially with no iteration).

But optimal planning can require scheduling information. For example, if a key resource will not be available during a certain period, then one may want to change the plan to take advantage of a different resource, possibly using a different process. Although scheduling usually cannot be done until a plan is established, in cases with highly constrained resources, the plan cannot be established until the schedule is (at least partially) established. In general, the more constrained the resources, the more tightly coupled planning and scheduling will be.

## **2.2. Stochastic-Resource Constrained Problem**

The stochastic resource constrained problem (SRCP) has been studied since the early 60s. In its most general form, the resource constrained project scheduling problem can be stated as: given a finite set of resources and a set of tasks, what is the best way to assign the resources to the tasks such that the objective is optimized? The general form assumes that 1) tasks are 'multi\_modal', i.e. the time, cost, quality, and any other performance measures of any task depends on the resource(s) assigned to it, 2) resources have temporal constraints \_ they are available in constant or variable quantities at specific times, 3) tasks have some degree of atomicity \_ there exists some discrete time interval into which each task can be divided, 4) the objective may have multiple criteria.

While many papers have been written describing various formulations, scheduling techniques, and optimization algorithms for the SRCP, it remains an area that is weak in reported research. Some of the first optimization formulations used integer programming. Although these have been proven to find optimal solutions, they are computationally intractable on problems of any significant size. Within the last 15 years, enumerative search techniques have been combined with heuristic scheduling rules to find optimal solutions to relatively small (5 resources, 50 tasks) projects with limited objective functions (typically minimize time\_to\_completion).

## **3. IMPLEMENTATION OF AN INTEGRATED WIND TUNNEL PLANNING SYSTEM: THOERY OF CONSTRAINTS**

One common problem with most planning and scheduling systems available to practitioners is that they give little guidance to project task start times, rather focusing on anticipative project duration times. Most systems treat project task times as deterministic events for planning purposes. Also, the availability of planning for resource constraints is limited. However, it is the stochastic nature of tasks combined with scarcity of resources that can cause a wide variation in overall project completion times. Allocation of scarce resources and selection of parallel task start task times can greatly influence this outcome. In the stochastic resource constrained project (SRCP) there are instances in which selection of one task to start before another can have drastic implications for stochastic task projects. Heuristic, simulation systems, and philosophical approaches have been developed to address

task times and resource allocation. One relatively new technique that has received attention is the *Theory of Constraints (TOC)*.

As part of this project, NASA LaRc and the Wind Tunnel Enterprise signed a contract with ProChain Solutions, Inc. for TOC project management consulting, education, and TOC scheduling software called the ProChain. The software has been successfully implemented during the project. ProChain Solutions will work with each wind tunnel operations group and the Wind Tunnel Enterprise will develop and implement TOC based schedules and project management controls.

Eliyahu Goldrat is credited with beginning the theory of constraints (TOC) methodology with his books *The Goal* (1989) and *Critical Chain* (1997). TOC is a common sense management philosophy that espouses that in order to improve the performance of any system, one must first find the constraint of the system and then concentrate effort on elevating the capacity of the constraint. This TOC overview is based on the current implementation of Goldrat's TOC methodology.

For simplicity's sake, in typical project management such as CPM, task durations are treated as if they are deterministic, when in fact they are highly probabilistic. Project managers are attempting to simplify their jobs using methodologies that were designed before the advent of computers, unknowingly causing many undesirable effects. Software allows a project manager to easily deal with the inherent uncertainty in project management and allow a schedule to be dynamic. Treating task duration estimates as probabilities solves many of the problems projects are currently facing using prevalent methodologies.

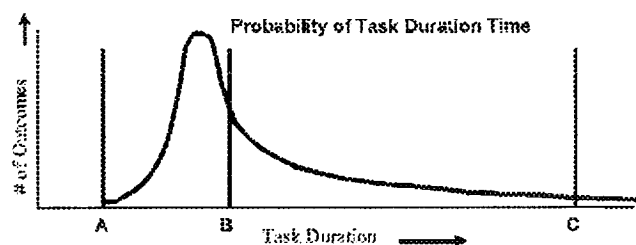


Figure 1. Distribution of Possible Task Duration Outcomes.

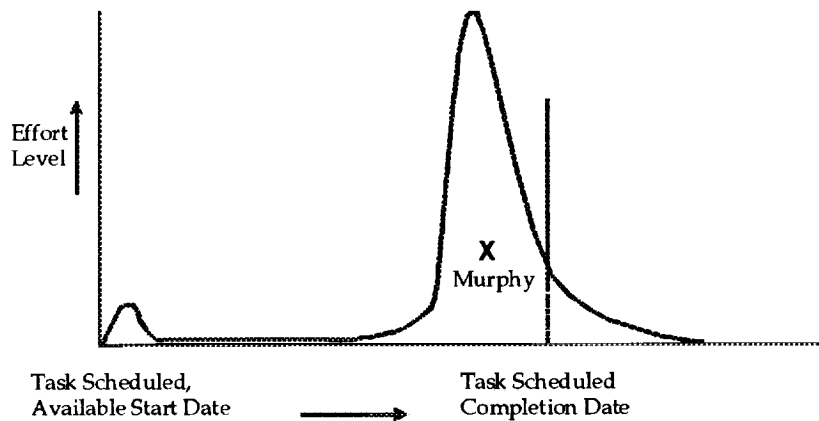
Goldratt contends that most schedules have safety time built into each task estimate. Figure 1 above shows a general distribution that represents the possible outcomes for most tasks. It is a normal distribution with a finite left tail because things can only go so well. A task will never be finished in

zero or negative time. On the other hand, quite a lot can go amiss. This explains the long right tail. When a person is asked to estimate task duration, consciously or subconsciously he or she is picturing a graph like the one above which is built from historical data points. Time “A” is the “Pure Success” basis. It is not likely to repeat and, unless someone is very new to project management, will never be given as a task estimate. Time “C” is highly achievable, even with a major disaster. This is the time commonly used because this is what incentive systems have trained workers to give as estimates. Goldratt calls this the “95% time” because 95% of the time a resource will finish in less time than the given estimate. Furthermore, if management has a practice of cutting all estimates across the board to squeeze a schedule into a given amount of time, this is even more incentive for team members to give “C” or even “C+” times. Time “B” represents the duration estimate such that 50% of the outcomes are less than it, and 50% of the outcomes are greater. In other words, if a resource gives this duration as an estimate he or she has a 50/50 chance of finishing on time. This is the median task duration and the time estimate one attempts to build into a Critical Chain schedule. Theoretically, the average task duration should be used, but studies have shown that even when people are asked to estimate an average duration, they instead provide the median duration. [Newbold, 1998] Because of the long right tail, the median duration should be slightly less than the average duration, but this difference should not be significant for CC’s purposes.

A portion of the time saved by moving from “C” to “B” task estimates is then aggregated at the end of the schedule to act as an overall “shock absorber” for the entire project. This aggregated buffer is called the “Project Buffer.” Consolidated buffer is also placed at the juncture of all non-critical paths with the project’s critical path. These buffers are called “Feeder Buffers” and their purpose is to protect the critical path from the variability of non-critical path tasks.

One problem with leaving small amounts of buffer time in each task estimate, instead of aggregating it, is that the safety is often wasted at the beginning of the task period, not the end where it can do good. Goldratt outlines three ways in which safety, or buffer, is typically wasted. The first is called “Student Syndrome.”

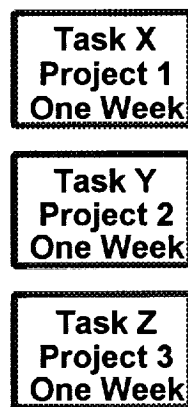




©Avraham Y. Goldratt Institute

Figure 2. Student Syndrome.

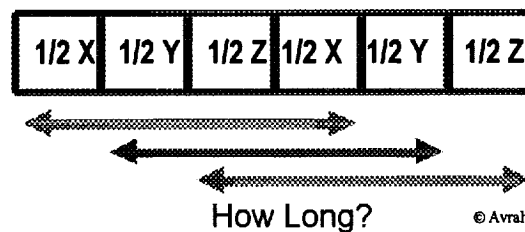
Goldratt writes that once a resource has negotiated a “C” time, they reevaluate the task and decide how long it will most likely take. Then they get caught up working on other projects with closer deadlines. When they have only the expected duration left until the deadline, they really ramp up the effort level. At that point, if they encounter an unexpected problem, the deadline is missed. Notice in Figure 2 that if the resource would have started the work when it was assigned, they still could have easily met the deadline, even with the “Murphy.”



In order to keep each project on track, a resource does half of task X, then half of task Y then half of task Z, then finishes task X then Y, then Z.

How long does each task take to complete?

What happened to the safety time?



© Avraham Y. Goldratt Institute

Figure 3. The Multiplying Effect of Multi-Tasking.

The next way in which safety is wasted is the multiplying effect of multi-tasking. If a resource has three tasks assigned in descending priority X, Y and Z, what often happens is that they will start task X and work on this unless the work is “reprioritized.” Most project managers are given the incentives that drive them to be concerned only with optimizing their project. Furthermore, they have learned from experience that in project management “the squeaky wheel gets the grease.” Thus, what will typically happen is that Y’s project manager will go to the resource and ask what progress he or she has made on his task. Not wanting to disappoint Y’s project manager, the resource will drop what he or she is doing on task X and start working on task Y. This same thing happens with task Z, leading to the workload shown in the Figure 3. What has happened is that each one-week task has taken at least two weeks to complete. This picture does not even show the set-up time required when switching between tasks. The net effect is that each task finishes later than it would have if our resource had worked on task X until it was complete, then task Y, and then task Z.

The last way in which safety is wasted has to do with the structure of most schedules. Because tasks can have multiple necessary predecessors, delays are passed on, while gains are not.

#### 4Figure 4: Schedule Structure Allows Delays to be Passed on while Gains are Not

The diagram and captions of Figure 4 illustrate how the structure of a conventional schedule causes a lose-lose situation. A delay in any of the three predecessors will be passed on to the successor, while any gains will not be passed on. Even if all three predecessors finish early, because we build our schedules around due dates, most likely the successor resource will not be ready to start his or her task early.

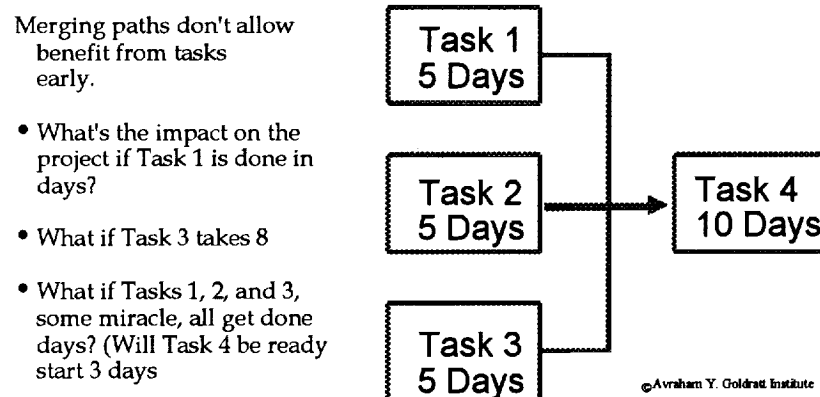


Figure 4. Schedule Structure Allows Delays to be passed on while Gains are not.

The next step is converting a Critical Path Schedule into a Critical Chain Schedule. This section uses a highly stylized PERT schedule with only two branches feeding into one final task. The simplicity of the schedule clarifies the logic of the transformation, which also scales up to schedules with complexity that is more realistic.

As can be seen in Figure 5, the letter in each block represents the resource (usually a person or group of people) necessary to complete the given task. Assume there is only one available resource for each given letter. The number in each block represents the estimated duration of the task in days.

The first step in the conversion is to attempt to pull the safety out of each duration estimate and get at the median times for each task ("B" times). In this example, this is shown as half of the initial "C" estimate. This is the rule of thumb that Goldratt recommends using.

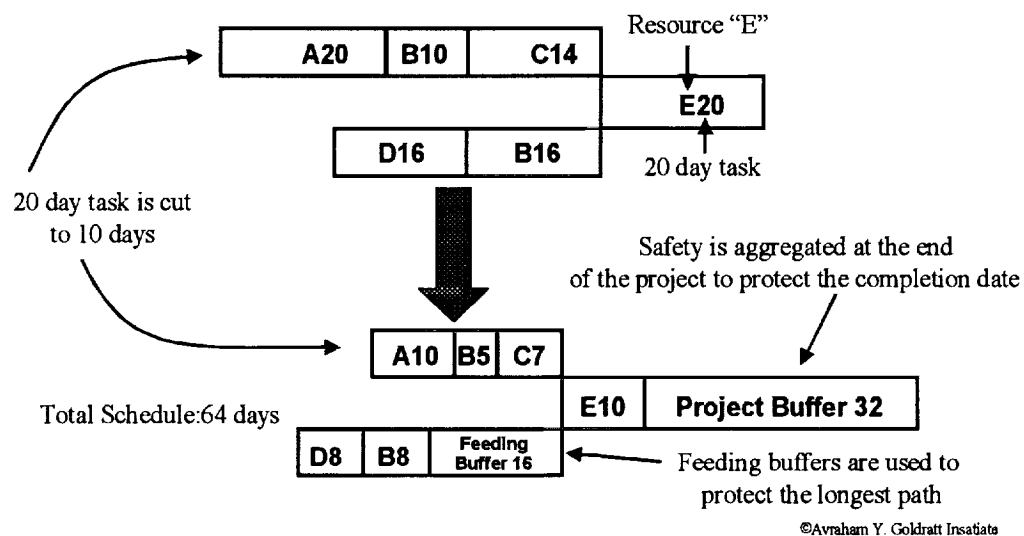


Figure 5. Aggregate the Buffer at the End of the Schedule.

The next step is to aggregate all of the safety at the end of the project where it can act as an overall buffer for the entire project. The date one would commit to a marketing campaign or to senior management would be at the end of this project buffer, not at the shown finish of task "E." Theoretically, this is the only date that remains fixed in a CC schedule. This project buffer becomes a necessary component of the schedule, and must not be considered "padding."

The safety from the shorter branch has also been consolidated into the feeding buffer. This serves to immunize the critical path against negative variations of the feeding path. In effect you are giving the feeding path its own safety so that if it runs into delays, unless the entire feeder buffer is “eaten,” the delays will not be passed on to the rest of the schedule.

In general with CC, one starts tasks as late as possible, which is a departure from how Microsoft Project will automatically schedule a project (it will start all tasks as early as possible). Because work is started later and generally worked on until it is complete, this helps reduce work in progress (WIP). From a management point of view this is beneficial because WIP is costly and can serve to bog workers down. This can also be a difficult paradigm shift for a team to make. Feeder paths are started a defined amount (feeder buffer) earlier than necessary as a risk reduction. As can be seen, at this point both versions of the schedule are 64 days long.

The next step is considered one of the most powerful aspects of Critical Chain. Because of random number aggregation theory, the overall variance of the critical chain will be much less than the addition of all the individual variances for each CC task. In other words, the amount of protection necessary when you aggregate all of the tasks is much less than if you added the protection originally built into each estimate. In the example below, the buffer is cut in half, leaving a buffer length of one half the length of the CC (Goldratt recommendation). The same applies to feeding paths, so that a feeding buffer is optimally one half the length of the feeding branch against which it is protecting.

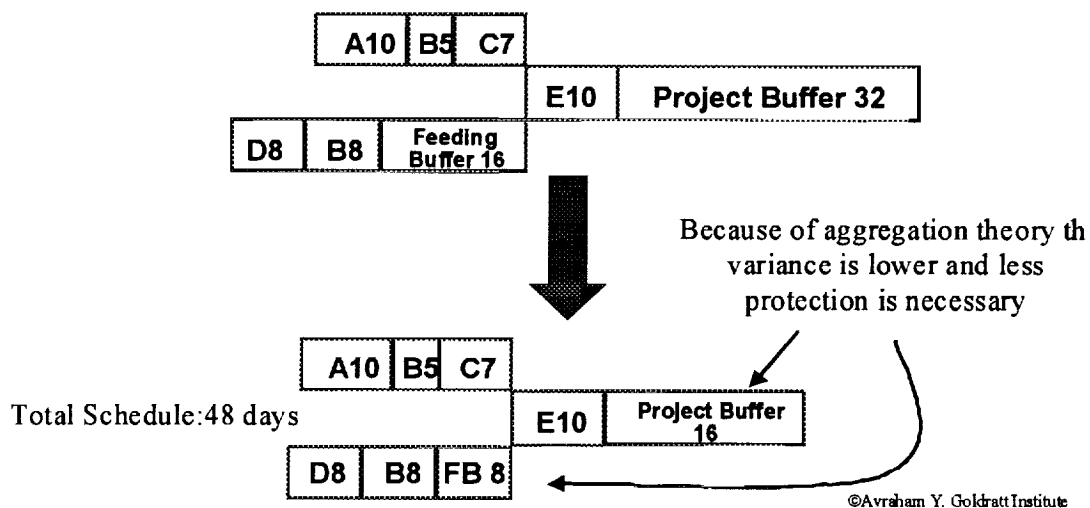


Figure 6. Aggregation Theory Allows Buffers to be Cut.

The schedule has been cut from 64 days to a total of 48 days (Figure 6), but Goldratt would contend this schedule has a much better chance of finishing on time than the original longer schedule. This is because the buffer has been concentrated at the end of the schedule where it can do good, allowing individual task early and late completions along the CC to counteract each other.

Notice that there is still one problem. Resource “B” is scheduled to perform two tasks simultaneously. Standard Microsoft Project and CPM will allow this, although most people recognize it leads to an unrealistic schedule. The next step in the conversion is to de-conflict resources. This is when the schedule is converted from a CPM schedule to a Critical Chain schedule.

As can be seen in Figure 7, the feeder branch is started earlier and becomes part of the longest chain of dependent tasks. The Critical Chain follows the dotted line from D8 to B8, then due to resource dependency up to B5. Notice that a feeder buffer is now placed in front of A10 to protect the CC against its uncertainty. Also, notice that the schedule has grown from 48 days to 57 days, but is much more realistic and predictable because resources are only counted upon to undertake one task at a time.

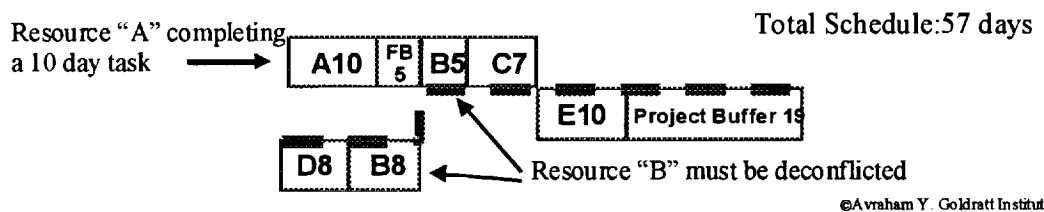
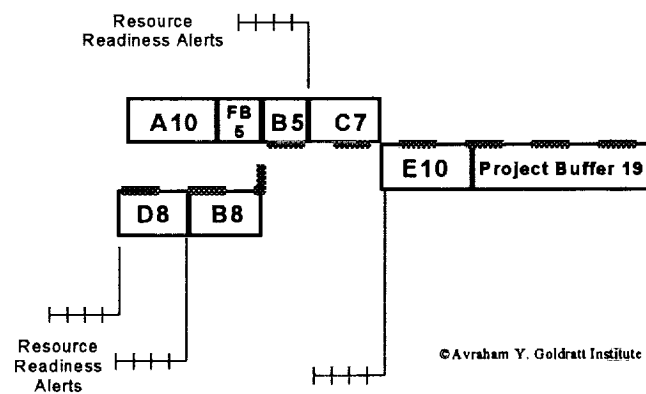


Figure 7. The Critical Chain Takes into Account Resource Dependencies.

In order to take advantage of early finishes and deal effectively with late finishes, a CC schedule must be very dynamic. Most project managers worry that logistically a dynamic schedule will become unmanageable, and thus opt for the “simpler” approach of trying to lock down the schedule by assigning due dates. While this might seem like a simpler approach, it leads to the undesirable effects mentioned earlier. The CC schedule does not become unmanageable because only the CC tasks must remain flexible. Unless there are major variances, feeder paths remain set.

The team makes a pact during the implementation of CC that all members will drop what they are doing to work on CC tasks and work non-stop on those tasks until they are complete. In this regard, a CC schedule is very similar to running a relay race. Team members wait to get the baton and then run as fast as they can until they pass the baton on to another teammate. The CC by definition defines the overall length of the project, thus it makes sense to concentrate all management attention on these tasks and “sprint” them as quickly as possible.

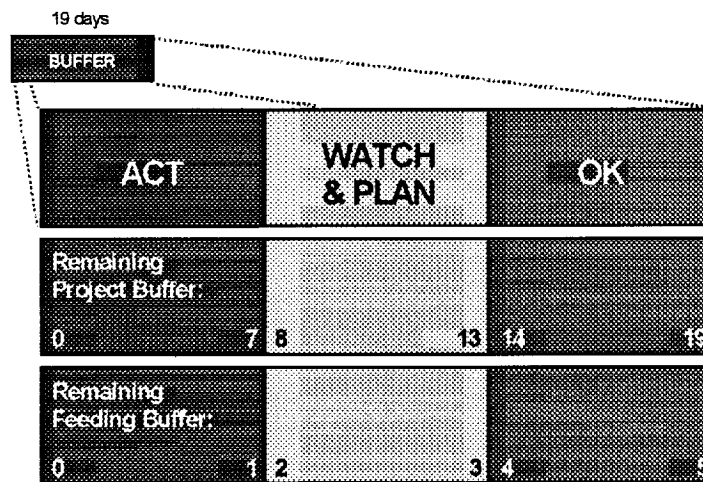
Resource Readiness Alerts (RRA's, called Resource Buffers by Goldratt) allow a project team to remain aware of the CC (Figure 8). RRA's are a form of communication between the schedule keeper and the rest of the team. The schedule keeper will get task updates (Goldratt recommends daily) from all resources currently working on tasks as the schedule progresses. The resources merely need to report how many workdays remain until they estimate their task is complete. When a predecessor CC resource reports to have five days remaining (Goldratt recommended), the schedule keeper informs the successor they have approximately five days until they are on the CC. This is a dynamic countdown for the successor. If the predecessor reported two days later that he or she hit a glitch and still had five days remaining, this would be passed on to the successor. This allows CC resources to plan their work schedules and keep other project managers aware of their pending CC status.



*Figure 0: Resource Readiness Alerts*

**Figure 8. Resource Readiness Alerts.**

The last major distinction of CC is the way in which the schedule is managed. Management controls the project by monitoring status of the buffers. This allows them to highlight the tasks that need immediate attention. A typical project will have numerous feeder buffers. The most consumed feeder buffer is protecting the feeder path with the highest probability of delaying the CC. Thus, management can focus attention on the feeder paths with the most depleted buffers. Of course, the CC tasks by definition are always considered crucial. The buffers also help management to act proactively. Buffer management highlights potential problems much earlier than they would ordinarily be discovered using typical project management techniques.



©Avraham Y. Goldratt Institute

*Figure 0: Buffers Provide Focus and Early Warning*

Figure 9. Buffer Management.

The feeder and project buffers are broken into three zones similar to a traffic light indicator. The “green” zone is labeled “OK.” If this portion is eaten into, management does nothing. This prevents micromanaging and over tweaking (this is very similar to SPC). The next zone of the buffer is labeled “Watch & Plan.” In this “yellow” zone, the team would form a plan to put into action if further buffer is eaten. The last zone is labeled “Act.” This is the “red” zone wherein management would expect its team to initiate the plans to increase the size of the buffer or at least prevent further erosion of it. Actions could be assigning more resources to certain tasks, working weekends, breaking finish-start constraints that might not have been necessary, etc. This is extremely powerful, especially

because management can set the size of each zone based on past performance, inherent variability of the project, etc. Setting the size of each zone will in some regards define the degree of management involvement in the project.

In Figure 9, the buffers are shown divided into three equal parts. Management should expect buffer to be eaten as the team advances through the project and runs into unexpected iterations or “Murphys.” Thus, the buffers need to be divided relative to how much CC is left in the project. If a project has nine months left to complete with only two weeks of Project Buffer remaining, there is a very high probability the due date will be missed. On the other hand, if a project has two weeks left of CC, and the same two weeks of buffer, management will most likely not be very concerned. Management needs to monitor the buffers relative to how much CC or feeder branch remains, and the rate of buffer consumption.

Within TOC, as with other scheduling problems, there are single and multi-project implementation issues. Here there are two main categories of CC. The first is called single project implementation. The definition of single project implementation is that the projects are not predominantly utilizing the same resources. Within an organization, there can be multiple single CC projects. The second type of implementation is called multi-project implementation. Multi-project implementation indicates there are multiple simultaneous CC projects and many of the resources are shared across projects. It is more difficult to schedule resources across multiple projects because there is no clear decision metric when deciding which project to first allocate a constrained resource. Goldratt contends that CC and TOC can provide this global project metric.

Multi-projects require a higher level of implementation with much higher coordination requirements. At this level, an organization identifies which resource, on average, is the constraint across all the projects it undertakes. More strategically, an organization would identify which resource should be its constraint, and then have this resource or group of resources pace when new projects are undertaken by the entire organization (Goldratt calls this resource the “drumbeat”). As an example, if electrical engineers were in very short supply and thus demanding a premium, a design organization might want to staff itself such that electrical engineers are its constraint. All other resources would have, on average, more design capacity so that the electrical engineers were always the bottleneck. Thus, management could plan the design workload such that electrical engineers were always at full capacity. As the electrical engineers completed tasks, they would “pull” more design work into the company.



Pulling more design work into the company than the electrical engineers could handle would only serve to increase work in progress (WIP). This would slow the organization and do nothing to increase its throughput. Electrical engineers would act as the “drumbeat” for the design organization much as a manufacturing bottleneck acts as the “drumbeat” for a plant. In this way, applying Goldratt’s Theory of Constraints (TOC) to project management is very similar to applying TOC to a factory, albeit the variances or uncertainties are typically much greater in project management.

Goldratt suggests that long term, most organizations should strive for the multi-project implementation. Since most of this research will be with single project implementation, the majority of this paper will discuss only single project implementation.

#### **4. CONCLUSIONS AND FURTHER RESEARCH**

In this study, an integrated project scheduling and management system based on Thoeory of Constraints (TOC) has been implemented for the wind tunnel operations at NAS Langley Research Center which will optimize the overall project duration. The TOC approach takes a philosophical approach to scheduling in an attempt to reduce time risk, address resource constraints, and provide the scheduling manager with realistic task strat times. However, within this philosophical approach there is a need for further study on scheduling optimization in an attemp to improve overall project duration. Current TOC project scheduling methods do not employ optimization techniques which attempt to reduce project time duration. The current system of task selection within TOC network construction uses an undefined selection methodology without an attempt for an optimum task order within the network. As further research, one needs to look at the integration of an optimization method with TOC in order to improve the overall project time duration while maintaining the inherent project risk component of the original TOC based network.